



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

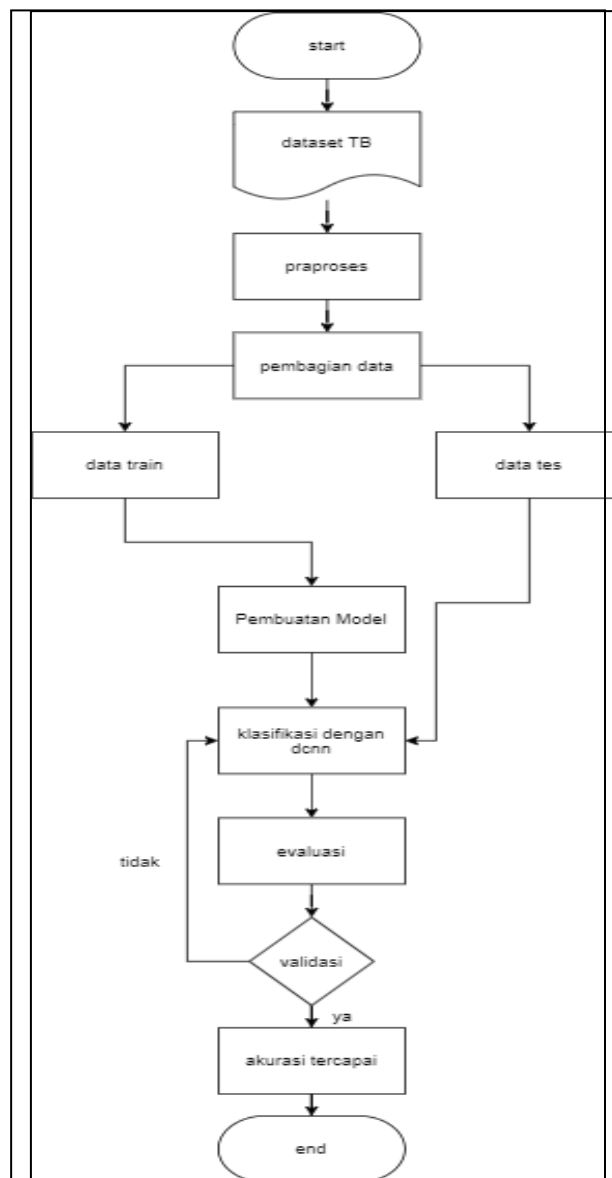
Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III METODOLOGI DAN PERANCANGAN SISTEM

3.1 Metodologi Penelitian

Berikut merupakan langkah-langkah yang dilakukan pada penelitian



Gambar 3.1 Flowchart Penelitian

3.1.1 Dataset TB

Data Gambar *rontgen* paru-paru merupakan *data* yang diperoleh dari Montgomery dan Shenzen (Jeager, dkk, 2014) (Candemir, dkk, 2014) pengambilan *data* dari Sampel *dataset* yang digunakan untuk riset ini adalah sebanyak 800 gambar *rontgen* yang berbeda. Selanjutnya gambar tersebut akan di *resize* menjadi ukuran 228 x 228 *pixel*, dilakukan *blurring* menggunakan *gaussian filter*, dan dilakukan *thresholding* menggunakan *otsu thresholding*.

3.1.2 Praproses

Praproses yang akan dilakukan pada *dataset* TB bertujuan untuk mempermudah proses gambar saat akan dilatih pada model dan mempercepat selesainya proses pelatihan *dataset* tersebut. Praproses yang dilakukan pada penelitian ini ada 4 yaitu: *grayscale* gambar, *resize* gambar, *gaussian filter*, dan *otsu thresholding*. Alur pada praproses dapat dilihat pada *flowchart* gambar 3.2.



Gambar 3.2 Flowchart Praproses

A. Grayscale

Grayscale merupakan suatu metode pengolahan gambar yang merubah gambar dari format RGB menjadi keabuan mengurangi dimensi yang dimiliki gambar tersebut. Untuk mengubah gambar RGB ke gambar grayscale diperlukan bobot intensitas yang berbeda untuk setiap warna. Berdasarkan fungsi matematis (2.1) bobot untuk warna merah adalah 0.33, bobot untuk warna hijau adalah 0.56, dan bobot untuk warna biru adalah 0.11.

B. Resize

Memperkecil ukuran gambar menjadi 228×228 *pixel* yang bertujuan untuk membuat gambar menjadi sama besar, memudahkan model untuk mempelajari fitur penting dari gambar tersebut lebih cepat dibandingkan dengan gambar yang lebih besar ukuran pixelnya sehingga mempercepat proses praproses selanjutnya dan pelatihan serta testing model.

C. Gaussian filter

Gaussian filter merupakan salah satu proses *image blurring* untuk mengurangi *noise* dari suatu gambar dan mempertajam tepi pada gambar. Hasil dari Gaussian blur merupakan angka dengan tipe data float. Untuk melakukan *Gaussian filter* diperlukan *array* (m, n) yang bernilai *integer* ganjil. Tujuan diterapkannya *gaussian filter* adalah untuk memperjelas tepi *dataset* gambar

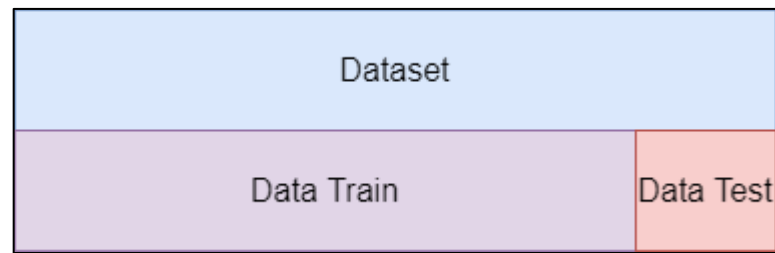
paru-paru.

D. Otsu Thresholding

Thresholding dilakukan untuk memisahkan obyek yang ada dalam suatu gambar dengan *background* gambar. Gambar yang diproses dapat memiliki nilai 0 hingga 255 untuk setiap *pixel*. Nilai-nilai pixel tersebut nantinya akan diubah menjadi bilangan biner. Metode *otsu thresholding* dipilih karena telah banyak digunakan dan memberikan hasil thresholding yang optimal.

3.1.3 Pembagian data

Setelah praproses dilakukan maka untuk melakukan suatu prediksi diperlukan pembagian *dataset* menjadi *data train* dan *data test*. *Data train* digunakan untuk melatih model yang akan dibuat sedangkan *data test* digunakan untuk validasi apakah model yang dibuat sudah baik atau belum. Pada penelitian ini *dataset* akan dibagi menjadi rasio 80% *data train* dan 20% *data tes* karena ratio 80 : 20 merupakan rasio yang sering digunakan dan ratio 80 : 20 juga berhubungan dengan *Pareto principle* yang menurut Koch (2018) 20% dari total data merupakan representasi dari 80% total data. Pada mesin learning 20% data testing merupakan representasi akurasi untuk validasi dari suatu model untuk memprediksi data yang baru Ilustrasi pembagian data ditunjuk pada gambar 3.3.



Gambar 3.3 Pembagian Data

Untuk melakukan pembagian data menjadi dua data yang sesuai dengan label data nya maka digunakan fungsi yang terdapat pada library sklearn, yaitu `train_test_split`. Fungsi ini memiliki lima parameter yang dapat mempengaruhi hasil pembagian data antara lain:

A. Test Size

Parameter yang berisi tipe data float yang nilainya berada di antara 0.0 hingga 1.0. Parameter ini merupakan representasi proporsi pembagian data menjadi *data test*.

B. Train Size

Parameter yang berisi tipe data float yang nilainya berada di antara 0.0 hingga 1.0. Parameter ini merupakan representasi proporsi pembagian data menjadi *data train*.

C. Random State

Parameter yang berisi tipe data integer yang akan memberikan angka *random* dalam pembagian data.

D. Shuffle

Parameter yang berisi tipe data Boolean. Jika berisi *false* maka parameter *stratify* harus kosong. Parameter ini akan mengacak *dataset* yang akan dibagi sebelum melakukan pembagian data.

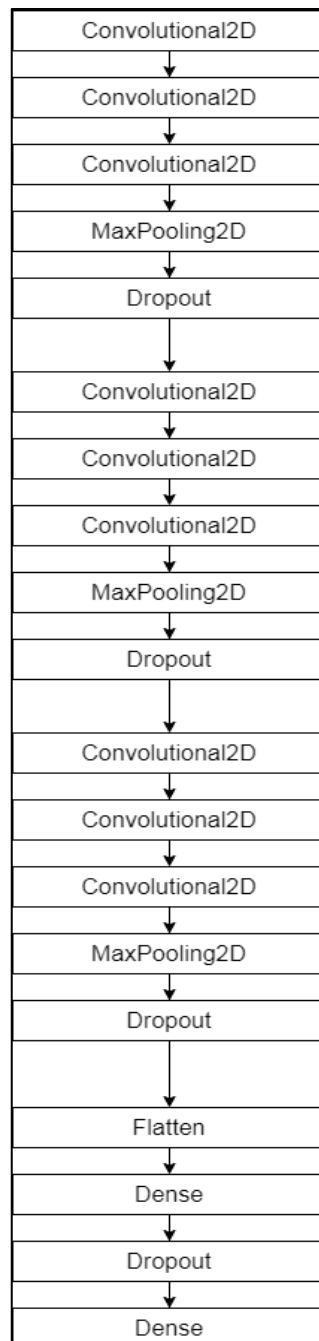
E. Stratify

Jika *stratify* tidak kosong, maka pembagian data dilakukan pengacakan sebagai satu kesatuan

Pada penelitian ini data gambar yang akan dibagi dimasukan kedalam variabel array X dan label untuk *dataset* kedalam variabel *array* y yang kemudian akan dimasukan kedalam fungsi `train_test_split()`. Parameter yang digunakan merupakan *test_size* sebesar 0.2 untuk membagi data menjadi 0.8 menjadi *data train* dan 0.2 menjadi *data test*. Pada penelitian ini tidak dilakukan K-fold cross validation karena dengan menggunakan k-fold cross validation proses pelatihan yang dilakukan akan menjadi k kali lebih lama dibanding dengan menggunakan `train_test_split` serta hasil testing dari `train_test_split` lebih mudah untuk di evaluasi secara mendalam.

3.1.4 Pembuatan Model

Proses pembuatan model akan dilakukan dengan pembuatan suatu variabel model. Gambaran pembentukan model DCNN ditunjuk pada gambar 3.4.



Gambar 3.4 Struktur Model DCNN

Model yang dibuat pada gambar 3.4 terdiri dari tiga bagian ekstraksi fitur yang didalamnya masing-masing terdapat tiga lapisan konvolusi dengan aktivasi relu, satu lapisan *pooling*, dan satu *dropout*. Bagian terakhir berisi satu flatten, satu *dense* dengan aktivasi relu, satu *dropout*, dan satu *dense* dengan aktivasi *softmax*.

3.1.5 Klasifikasi Dengan DCNN

Setelah model dibuat, model akan di latih dengan *data train* yang telah didapatkan dari proses pembagian data dengan menggunakan fungsi *fit* yang terdapat pada *library* keras. Untuk dapat menggunakan fungsi ini dibutuhkan beberapa parameter seperti data dan label *data train*, jumlah *epoch*, data dan label *data test*.

Jumlah *epoch* yang digunakan untuk melatih model ini adalah 100 *epoch*. Hasil model yang setelah selesai dilatih akan disimpan dalam format pickle (*.pkl) yang digunakan untuk prediksi gambar.

3.1.6 Evaluasi

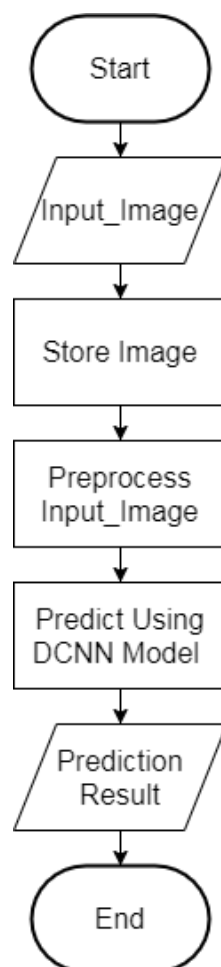
Melakukan evaluasi pada model yang telah dibuat. Metode yang digunakan untuk melakukan evaluasi ini adalah dengan melihat hasil dari *confusion matrix*. Hasil dari *confusion matrix* akan digunakan untuk mencari nilai sensitifitas dan spesifisitas.

3.2 Perancangan Aplikasi

Aplikasi diagnosa penyakit tuberkulosis paru-paru dengan algoritma deep convolutional neural network memiliki perancangan dalam membangun aplikasi. Perancangan sistem yang digunakan merupakan flowchart dan desain antarmuka.

3.2.1 Flowchart

Berikut merupakan flowchart untuk perancangan aplikasi berbasis website untuk diagnosa penyakit tuberkulosis.



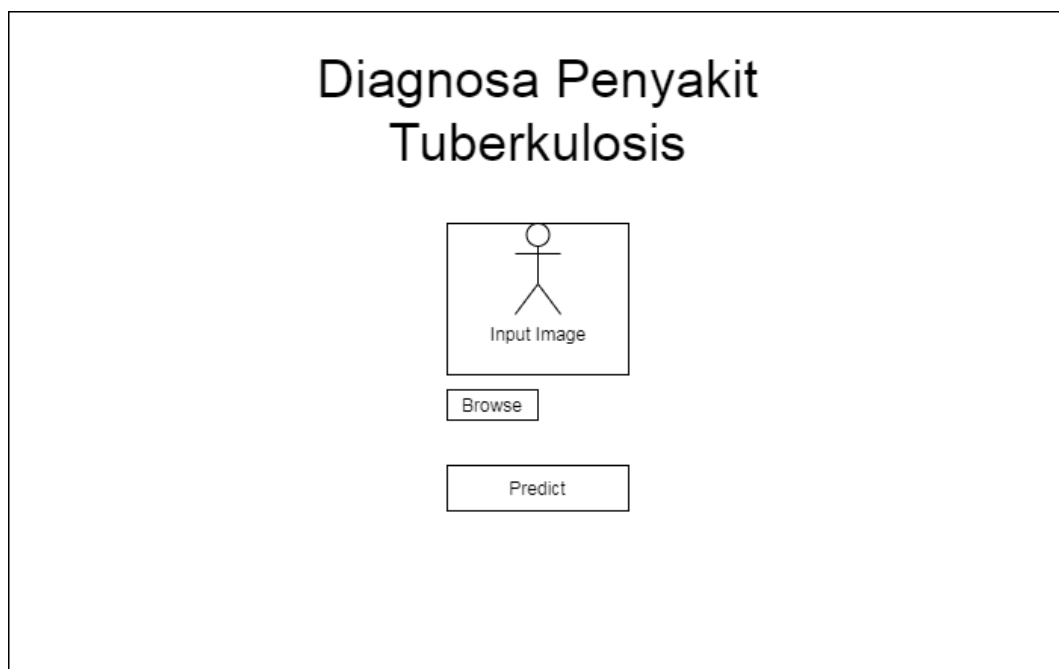
Gambar 3.5 Flowchart Aplikasi

Gambar 3.5 merupakan flowchart perancangan aplikasi. Pada aplikasi ini pengguna memasukan gambar paru-paru yang ingin di diagnosa. Aplikasi akan menyimpan gambar tersebut kedalam folder Upload lalu gambar dipraproses.

Aplikasi akan melakukan prediksi dengan model yang telah dibuat terhadap gambar yang telah dipraproses. Jika prediksi sudah selesai maka akan ditampilkan diagnosa terhadap gambar tersebut serta akurasi prediksi yang dilakukan.

3.2.2. Desain Antarmuka

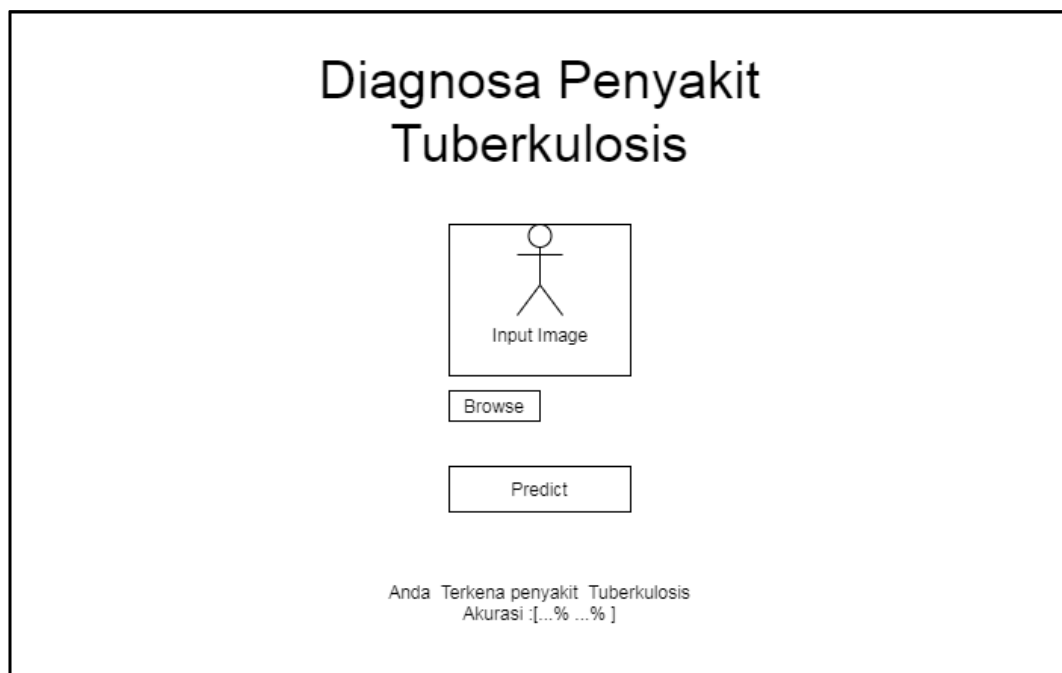
Desain antarmuka aplikasi digunakan untuk memberikan gambaran tampilan dari aplikasi yang akan dibuat.



Gambar 3.6 Desain Antarmuka aplikasi

Gambar 3.6 merupakan desain antarmuka dari halaman aplikasi.

Pengguna mendapatkan tampilan ini ketika pengguna membuka aplikasi. Pada halaman ini pengguna dapat memberikan input berupa gambar dengan menekan tombol browse. Gambar yang diinput akan di tampilkan pada kotak diatas tombol browse. Ketika pengguna menekan tombol predict maka gambar yang diinput tersebut akan tersimpan dan dilakukan proses prediksi yang hasilnya akan ditampilkan pada gambar 3.7.



Gambar 3.7 Desain Antarmuka Aplikasi Setelah Input

Kemudian jika pengguna ingin melakukan prediksi selanjutnya, pengguna bisa langsung memasukan gambar baru dengan menekan tombol brow